# WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Monday, March 27, 2006

| Hide? | Set Name | Query | Hit Count |
|---|---|---|---|
| | *DB=PGPB; PLUR=YES; OP=ADJ* | | |
| ☐ | L14 | L13 and (authentication or authorization) | 1 |
| ☐ | L13 | L12 and (user profile) | 1 |
| ☐ | L12 | 20030163513.pn. | 1 |
| ☐ | L11 | L10 and (authentication or authorization) | 1 |
| ☐ | L10 | L9 and billing | 1 |
| ☐ | L9 | L7 and (user profile) | 1 |
| ☐ | L8 | L7 and profile | 1 |
| ☐ | L7 | 20030135628.pn. | 1 |
| ☐ | L6 | 2003/0135628.pn. | 0 |
| ☐ | L5 | l3 and user | 1 |
| ☐ | L4 | L3 and profile | 0 |
| ☐ | L3 | 20030055624 .pn. | 1 |
| ☐ | L2 | ('us20030055624')!.PN. | 0 |
| ☐ | L1 | ('us20030055624')!.PN. | 0 |

END OF SEARCH HISTORY

Previous Doc Next Doc Go to Doc#
First Hit

☐ Generate Collection

L3: Entry 1 of 4 File: PGPB May 26, 2005

DOCUMENT-IDENTIFIER: US 20050114701 A1
TITLE: Federated identity management within a distributed portal server

Detail Description Paragraph:
[0070] Commonly-assigned and co-pending U.S. patent application 20030135628 (Ser. No. 10/047,811; attorney docket RSW920030199US1), which is titled "Provisioning Aggregated Services in a Distnbuted Computing Environment", discloses techniques that enable heterogeneous identity systems to be joined in the dynamic, run-time Web services integration environment. This application, referred to herein as "the provisioning invention", is hereby incorporated herein by reference. A provisioning interface was disclosed in the provisioning invention to enable automatically and dynamically federating the heterogeneous identity systems which may be in use among the services which are aggregated as a composite service. The techniques disclosed therein allow users (whether human or programmatic) to be seamlessly authenticated and authorized, or "identified", for using the dynamically-integrated services. According to the provisioning invention, this seamless identification may be provided using a single sign-on, or "unifed login", for an aggregated service, wherein the provisioning interface of the aggregated service can be used to solicit all required information from a user at the outset of executing the aggregated service. A "stacking" approach was described whereby user passwords (or other credentials, equivalently, such as tickets or digital certificates) to be provided to the sub-services of an aggregated service are encrypted for securely storing. The sub-services are invoked in a specified order during execution, according to a definition that is preferably specified in the Web Services Flow Language ("WSFL"), and the stacked passwords are then unstacked and presented to the appropriate authentication or authorization sub-service.

Detail Description Paragraph:
[0072] The provisioning invention discussed publishing the provisioning interface, for each sub-service of an aggregated service, to a network-accessible registry using a WSDL document, thereby enabling the joining of identity systems for (sub-) services which are dynamically integrated. The provisioning invention also stated that the provisioning interface of the aggregated service can then be created by manually or programmatically selecting from the interfaces of the sub-services comprising the aggregation, and a WSDL document may be created for this new provisioning interface and published, in a recursive manner.

Previous Doc Next Doc Go to Doc#

**End of Result Set**



L11: Entry 1 of 1 File: PGPB Jul 17, 2003

DOCUMENT-IDENTIFIER: US 20030135628 A1
TITLE: Provisioning aggregated services in a distributed computing environment

Abstract Paragraph:
Methods, systems, and computer program products are disclosed for provisioning software resources used with aggregated web services. The disclosed techniques enable heterogeneous identity systems to be joined in the dynamic, run-time web services integration environment. Authentication and authorization may now be performed for the aggregated service, as well as for its sub-services. SOAP ("Simple Object Access Protocol") messages, as an example, may be used to relay identity information among distributed services, whereby credentials may be specified in the SOAP message header to accompany a service request specified in the SOAP message body.

Pre-Grant Publication (PGPub) Document Number:
20030135628

Summary of Invention Paragraph:
[0012] In particular, consider that many application services which are provided in a conventional manner require users to be authenticated and authorized before using those services. Authentication in this context means determining that the user is in fact who he purports to be, and authorization typically means determining what this user's access privileges are or whether this user is allowed to access a particular service or function thereof. In the web services environment, the intent is that a service provider may be located dynamically to perform a particular service. If multiple service providers are available, a particular one of these service providers might be selected based upon criteria such as the price of using this provider's service, response time guarantees of this provider's service, and so forth. It is possible that each provider might have differing formats for authentication and authorization information, as well as unique ways to access the authentication and authorization functions. There are no techniques known to the present inventors for federating, or joining, heterogeneous identity systems in the web services environment, which will be a serious inhibitor to use of aggregated web services.

Summary of Invention Paragraph:
[0015] A further object of the present invention is to define techniques for allowing users to conveniently access dynamically-located web services which require authentication and authorization.

Summary of Invention Paragraph:
[0021] In preferred embodiments, the analyses comprises at least one of (1) authentication and (2) authorization of the credentials.

Detail Description Paragraph:
[0043] The system interface is used for run-time management of portlets (that is, of web services represented by portlet proxies) by the portal platform Use of the system interface allows the portal platform to perform functions such as logging of

events, billing, and other types of administrative operations pertaining to execution of the web service. Two-way communication between the portal platform and the portlet proxy is used for this purpose.

Detail Description Paragraph:
[0051] A developer who creates the source code for a software resource to be deployed as a web service specifies the authentication, authorization, and/or configuration methods to be provided by that service. The services may then be aggregated as described in the related inventions, and the techniques of the present invention may be used for provisioning the aggregated service. For example, suppose the aggregated service is designed to provide e-mail services for a human user. A sub-service may be provided to establish a user's e-mail account. Typically, this account establishment sub-service will need input information such as the user's full name, an e-mail user identifier to be associated with this person, a password with which this person will access his email account, and perhaps configuration information such as how much storage should be allocated for this user's e-mail messages. (The stored password can be used subsequently, in combination with the user identifier, to authenticate this user as he accesses his e-mail messages using another sub-service of the aggregated e-mail service.) Access rights information might also be provided as input to the account establishment sub-service. A user who is a systems administrator, for example, might be given additional access rights for performing operations such as increasing the storage space allocation of another user, deleting the e-mail of another user, and so forth. WSDL documents may then be used to define the operations provided by each sub-service, and the messages and parameters which are used to invoke those operations.

Detail Description Paragraph:
[0054] Unified authentication and authorization operations are made more difficult by the dynamic nature of both the discovery and invocation of distributed services. The techniques disclosed herein address this difficulty by enabling an aggregated service to be provisioned within the context of a web services work flow, where operations are identified using WSDL documents and are invoked using SOAP messages within a work flow definition.

Detail Description Paragraph:
[0055] Aggregated services may constrain access to their exposed operations to those users who have sufficient credentials, and successfully demonstrate these credentials using an exposed authorization operation. It may also be advantageous to enable creation of user profiles which span an aggregated services, and optionally to allow these user profiles to be queried, changed, and/or deleted using corresponding service operations.

Detail Description Paragraph:
[0057] The "InResolveProvisioningIDRequest" message 502 illustrates an input request message which may be used to query a service for its view of who a particular authenticated user or entity is. (Hereinafter, the term "user" may be construed as applying equivalently to a human user or a programmatic entity such as an automated service unless specifically qualified.) Message specification 502 declares that this request takes a parameter named "authToken", which is a string type. For example, suppose a human user has been authenticated to an aggregated service, and that the aggregated service is holding an authentication token "X" for that human user. Further suppose that the aggregated service wishes to programmatically determine how this human user is known to a particular sub-service "ServiceABC". The aggregated service needs to locate a provisioning system which has information about that user. Messages 502 and 504 may be used to provide this functionality, where the token "X" is passed to the "ResolveProvisioningID" operation of "ServiceABC" (preferably, using a SOAP message, as will be described with reference to FIGS. 7A and 7B). As shown in FIG. 5D, "ResolveProvisioningID" 552 is an operation having an "InResolveProvisioningIDRequest" message (see element

502 of FIG. 5A) as well as an "OutResolveProvisioningIDResponse" message (see element 504 of FIG. 5A). The "OutResolveProvisioningIDResponse" message 504 is defined as returning a parameter named "Identifier" (of string type). Preferably, the returned identifier is an identifier of the remote provisioning system. This identifier may then be used as an input parameter for subsequent operations (see messages 506, 510, and 526, for example, which are described below), to specify the provisioning system which is managing the user profile or service configuration information, as the case may be.

Detail Description Paragraph:
[0058] Referring now to FIGS. 7A and 7B, preferred embodiments of the present invention use SOAP messages for communication among web services. The example SOAP message 700 comprises a SOAP envelope carrying a digital signature in its header, according to the prior art. See FIG. 7A for the header 710 and digital signature 720. This digital signature may be used for authentication of the requester who submits the service request carried in the SOAP message body. See FIG. 7B for the message body 730 and request 740. In this sample message 700, the message body specifies a "GetLastTradePrice" message, for which the <m:symbol>child element has a value of "IBM". It can be presumed that this an invocation of a stock quote service, and that this service requires the user to be authenticated; the digital signature of the user has therefore been supplied in the SOAP header. (Refer to "SOAP Security Extensions: Digital Signature, W3C NOTE February 6, 2001", which may be found on the Internet at location http://www.w3.org/TR/SOAP-dsig/, for more information about using SOAP messages in this manner.)

Detail Description Paragraph:
[0059] The present invention leverages this digital signature technique for conveying authentication information pertaining to authenticating users of aggregated web services, determining authorization of those users, and/or configuring aggregated web services.

Detail Description Paragraph:
[0060] Returning to the discussion of the sample provisioning interface messages in FIG. 5A, the "InResolveUsersRequest" message 506 illustrates an input request message which may be used to determine the set of users who are authorized to access a particular service. In the example, an authentication token is passed to the service being queried, and in this message, preferably serves to authenticate the information requester (that is, the programmatic entity or human user who is requesting the authorized users information). The "provID" parameter may be used to provide an address (such as a Uniform Resource Identifier, or "URI") of a provisioning system hosted by a service provider. The "ResolveUsers" operation (see element 554 of FIG. 5D) of a service receives the "InResolveUsersRequest" message 506, and responds with an "OutResolveUsersResponse" message 508. In the example, this output message 508 is defined as returning an array named "UserSet". The syntax "SOAP-ENC" in the part element of message 508 is a namespace prefix, and is used to qualify the array definition. (This output array presumably identifies the authorized users of the particular service hosting this "ResolveUsers" operation 554, which was bound to using UDDI and invoked using a SOAP message. As the "ResolveUsers" operation executed, it may have requested a provisioning system to perform the determination of authorized users.)

Detail Description Paragraph:
[0061] The "InCreateUserProfileRequest" message 510 shows how the interface of an input request message that creates a user profile might be designed. As in the other example messages, it is beneficial to include an authentication token as one of the input parameters passed to the remote service, so that the remote service can authenticate the information requester and determine whether this requester to authorized to use the "CreateUserProfile" 556 service which exposes the "InCreateUserProfileRequest" message 510. The "provID" parameter may be used to provide a URI or other address of a provisioning system, as discussed above, where

the user's profile is to be stored in this provisioning system. The "userID" parameter preferably identifies the user for whom (in the case of a human user) or for which (in the case of a programmatic user) the profile is being created. A "password" parameter may be provided to establish the password associated with this user. (Credentials other than a password might be used for this purpose, if desired.) The user's full name might be passed in a "FullName" parameter, depending on the needs of the underlying service. Finally, in this sample message, the user's access rights are provided as an array. The "CreateUserProfile" operation 556 receives the "InCreateUserProfileReques- t" message 510, and responds with an "OutCreateUserProfileResponse" message 512. In the example, this output message 512 returns a Boolean value indicating whether the profile creation was successful or not.

Detail Description Paragraph:
[0062] The "InQueryUserProfileRequest" message 514 shows an example interface for an input request message that is used to retrieve information from a user's previously-stored profile. The message parameters include an authentication token "authToken" for authenticating the information requester, a provisioning identifier "provID" for identifying a provisioning system " where the profile is stored, and a user identifier "userID" to identify the user for whom/which the profile information is being requested. This message 514 is provided as the input interface to a "QueryUserProfile" 558 service, and the "OutQueryUserProfileResponse" message 516 of this example returns the user's password, fill name, and access rights from the stored profile.

Detail Description Paragraph:
[0064] The "InDeleteUserProfileRequest" message 522 and "OutDeleteUserProfileResponse" message 524 are provided as the input and output interface of the "DeleteUserProfile" operation 562 (see FIG. 5E), and enable deleting a user's profile in a similar manner to how the profile may be created or updated with the "CreateUserProfile" operation 556 and "UpdateUserProfile" operation 560.

Detail Description Paragraph:
[0065] In addition to authentication and authorization messages such as those which have been described, it may also be useful to define messages and operations pertaining to configuration of aggregated web services. Examples of the "SetConfigParameter" 564 and "GetConfigParameter" 566 operations are illustrated in FIG. 5E.

Detail Description Paragraph:
[0066] The sample input message for the "SetConfigParameter" 564 operation is "InSetConfigParameterRequest" 526, and the sample output message is "OutSetConfigParameterResponse" 528. The input message 526 in the example has input parameters which include the authentication token "authToken" for the requester, the provisioning identifier "proviD" to identify the provisioning system where the parameter value should be stored, the user identifier "userID" to identify the user with whom/which this parameter should be associated, and the configuration parameter's name "parametername" and value "parameterValue". The output message 528 returns a Boolean value "result", indicating whether the "SetConfigParameter" operation succeeded.

Detail Description Paragraph:
[0070] The operations which are defined sequentially within the WSFL work flow of an aggregated service are executed, according to the work flow definition. The login information obtained from the user is preferably "stacked" for use by the sub-service to which individual elements of the login information pertain. Stacking of modules is known in the art by those familiar with identity systems and authentication systems which provide single sign-on capability. Stacking refers to using a "primary" password as an encryption key, where the information thus

encrypted comprises one or more "secondary" passwords. As the stacking process is used with the present invention, the secondary passwords are the passwords used for the sub-services, and the primary password applies to the scope of the aggregated service and protects these secondary passwords. The sub-services are invoked in a specified order, according to the WSFL definition, and the stacked passwords are then unstacked and presented to the appropriate authentication or authorization sub-service.

Detail Description Paragraph:
[0071] This process begins at Block 600 of FIG. 6, where the user identifier and password (or similar type of authentication input) are obtained. (Note that references herein to "passwords" are not meant to limit the type of credentials that may be supported. Credentials may be provided in many ways, including clear text, strings which have been encrypted, tickets, and public key security certificates such as X.509 certificates.) This authentication information may then be passed as input to a remote service, which will generate an authentication token (Block 610) upon invocation of its authentication operation.

Detail Description Paragraph:
[0072] Preferably, the authentication token generated in Block 610 is generated as an XML fragment, which can then be included in a SOAP message header. In this manner, user identities may be relayed when accessing web services. Refer to the discussion of the sample SOAP message 700 in FIGS. 7A and 7B, which shows how a digital signature is included in a SOAP header using XML syntax. (As shown therein, the digital signature tokens use a qualified namespace, and are therefore preceded by the letters "ds".) Authentication systems and policy systems may be bound to service operations using the SOAP header as well. WSDL descriptions preferably model operations as a combination of a SOAP header and body. That is, all operations requiring proof of identity preferably require user credentials to be exchanged. The SOAP Security Extensions technique used in the examples herein is one example of how this may be accomplished. The Security Association Markup Language ("SAML"), the Generic Security Service ("GSS") API, and the Common Secure Interoperability ("CSI") architecture also provide means for security exchanging a principal's credentials. (A version of SAML is defined in an OASIS Draft which may be found on the Internet at http://www.oasis-open.o-rg/committees/security/docs/draft-sstc-saml-spec-00.PDF, dated Apr. 11, 2001. The GSS-API is defined in RFC 2743, "Generic Security Service Application Program Interface, Version 2, Update 1", dated January 2000. CSI is defined in "Common Secure Interoperability V2 Specification", available on the Internet at http://www.omg.org/cgi-bin/doc?ptc/2001-03-0- 2.)

Detail Description Paragraph:
[0073] The token generated at Block 610, using the input information obtained in Block 600, is referred to herein as a "general" authentication token in that it preferably serves as a surrogate for this user which can be used subsequently to identify the user to various sub-services of the aggregated service. (In other words, this token is preferably not specific to any one sub-service or operation.)

Detail Description Paragraph:
[0075] In Block 625, the stacked identity information for the next operation to be performed is retrieved. This retrieved information is passed to this next operation's authentication service, which generates (or retrieves) an operation-specific token using this identity information.

Detail Description Paragraph:
[0076] At Block 660, the operation-specific token is returned to the caller using a SOAP header (as described with reference to FIGS. 7A and 7B). (Note that while the response messages in FIGS. 5A through 5C do not illustrate returning authentication tokens, such tokens can be added if desired.) Block 670 then uses the received operation-specific token to determine the user's operation-specific authorization.

(Users may have a number of roles which determine their credentials for a specific class of operations. A person who is a manager might be allowed to view the personnel records of his employees when acting in his manager role, as one example, whereas he might not be allowed to use this same operation to see his own personnel record when acting in his role of an employee.) The authorization invocation in Block 670 preferably also uses a SOAP header, for passing the operation-specific token received in Block 660. If the result of the authorization operation indicates that the user is authorized for the next operation to be performed in the aggregated service, then processing proceeds at Block 680. (Otherwise, an error may be generated and/or the flow might proceed to a different operation. The particular processing may vary from one implementation to another, and thus has not been illustrated in FIG. 6. It will be obvious to one of ordinary skill in the art how appropriate logic may be added to FIG. 6.)

Detail Description Paragraph:
[0077] Block 680 invokes the next sequential operation. This invocation may also use a SOAP header, if user credentials are required, for passing the operation-specific token received in Block 660. (If an authorization token is received as a result of the processing of block 670, that token may be passed in addition to or instead of the token from Block 650.) After the operation completes, Block 690 checks to see if there are more operations in the sequence. If not, then the processing of FIG. 6 ends. Otherwise, control returns to Block 620 to determine if the user is still authenticated for the aggregated service (after which Block 630 will determine whether the user is authenticated for this next service, as has been discussed earlier).

CLAIMS:

6. The method according to claim 4, wherein the analyzing step comprises at least one of (1) authentication and (2) authorization of the credentials.

Previous Doc Next Doc Go to Doc#

L14: Entry 1 of 1 File: PGPB Aug 28, 2003

DOCUMENT-IDENTIFIER: US 20030163513 A1
TITLE: Providing role-based views from business web portals

Abstract Paragraph:
Methods, systems, and computer program products are disclosed for providing role-specific views from a business web portal which supports one or more aggregated web services, where a "business web portal" is a collection of one or more portals which may be hosted by potentially disparate, autonomous service providers. This may be useful, for example, to extend the services of a particular business by programmatically including services of other enterprises. The disclosed techniques enable heterogeneous user profiles to be federated and exchanged in the dynamic, run-time web services integration environment. In this manner, users having particular roles can be programmatically presented with different views into an aggregated service. XML Linking language ("XLink") is preferably used to associate role-specific views of a particular sub-service from the aggregation with the role (s) to which that view pertains.

Pre-Grant Publication (PGPub) Document Number:
20030163513

Summary of Invention Paragraph:
[0022] Preferably, the user role is stored in a user profile associated with the user, and the user role is determined using the user's identification and credentials.

Summary of Invention Paragraph:
[0023] The technique may further comprise programmatically relaying user role information (which may include additional user profile information) among distributed services performed by the software resources of the aggregated service. In this case, the programmatic relaying may comprise sending a message which specifies the user role in a header of the message and in which a body of the message identifies that this message is delivering the user role. The message is preferably a SOAP message.

Detail Description Paragraph:
[0043] It is expected by the present inventors that significant advantages can be realized by providing role-specific views into the value chains, where the multiple views will be based on the services and/or information which are relevant to a particular role. The present invention is directed toward providing these role-specific views for aggregated services. It will be appreciated by those familiar with the art that participants may be authenticated by disparate security systems which are not managed by a central authority. Thus, it is assumed that both authentication and credential acquisition occurs in a federated manner. Referring again to the shopping example, along with the illustration provided in FIG. 2, users 240 who have the role of administrator might be allowed to create a composite shopping service, and to add or delete services from the composite service. For example, the administrator might add a customer feedback sub-service (not shown in FIG. 2). Users 220 who have the role of consumer, on the other hand, might be

provided a view which limits them to browsing items which can be purchased and placing orders--and which perhaps gives them access to information about their previously-placed orders. Users 270 with a role such as "business management" might be allowed to make various types of changes to the composite service (or to its sub-services), such as selecting the providers of the sub-services, changing prices of items offered for sale, modifying delivery agreements or other types of trading partner agreements, and so forth.

Detail Description Paragraph:
[0047] To provide role-based views for business webs which may integrate services from a number of different sources, it is necessary to be able to automatically and dynamically "federate" or join the heterogeneous user profile information they may use. The exchange of profile information must be done in real time so that user roles can be seamlessly determined, and an appropriate view can be presented which aggregates content appropriate for that role. Furthermore, it is desirable to provide this user profile information using a single sign-on approach, whereby identifying information obtained when a user begins to use a portal can be programmatically obtained and used by sub-services of an aggregated service, because requiring users to identify themselves repeatedly during the course of a particular service would likely cause user frustration and would be time-consuming and inefficient. The present invention provides a solution for these requirements, and leverages a number of open industry standard technologies in doing so, as will be described.

Detail Description Paragraph:
[0048] As used herein, the term "federated profile exchange" refers to a process whereby a federation authentication of an end user is performed (as disclosed in the provisioning invention); security attributes (such as the user's role) which are relevant for authorization are acquired, for this authenticated user; and profile data associated with these security attributes is resolved.

Detail Description Paragraph:
[0059] The provisioning interface disclosed in the provisioning invention enables automatically and dynamically federating the heterogeneous identity systems which may be in use among the services which are aggregated as a composite service. The techniques disclosed therein allow users (whether human or programmatic) to be seamlessly authenticated and authorized, or "identified", for using the dynamically-integrated services. This seamless identification may be provided using a single sign-on, or "unified login", for an aggregated service, wherein the provisioning interface of the aggregated service can be used to solicit all required information from a user at the outset of executing the aggregated service. (However, it may happen that some information needs to be requested from the user during execution, and in this case, use of the provisioning invention enables minimizing such requests.) A "stacking" approach was described whereby user passwords (or other credentials, equivalently, such as tickets or digital certificates) to be provided to the sub-services of an aggregated service are encrypted for securely storing. The sub-services are invoked in a specified order during execution, according to the WSFL definition, and the stacked passwords are then unstacked and presented to the appropriate authentication or authorization sub-service.

Detail Description Paragraph:
[0065] The techniques disclosed in the provisioning invention address the difficulty of providing unified authentication and authorization by enabling an aggregated service to be provisioned within the context of a web services work flow, where operations are identified using WSDL documents and are invoked using SOAP messages within a work flow definition.

Detail Description Paragraph:
[0066] The provisioning invention discussed the fact that aggregated services may

constrain access to their exposed operations to those users who have sufficient credentials, and who successfully demonstrate these credentials using an exposed authorization operation. The provisioning invention also stated that it may be advantageous to enable creation of user profiles which span an aggregated service, and optionally to allow these user profiles to be queried, changed, and/or deleted using corresponding service operations of the provisioning interface. The aggregated service may also be configured using information obtained with the provisioning interface, as stated therein, and user profiles may include user access rights information. One use of user rights which was briefly discussed in the provisioning invention is to determine the user's operation-specific authorization. For example, users may have a number of roles which determine their credentials for a specific class of operations. A person who is a manager might be allowed to view the personnel records of his employees when acting in his manager role, as one example, whereas he might not be allowed to use this same operation to see his own personnel record when acting in his role of an employee. The discussion of views based on roles was limited to this data-specific access-restriction example, and did not describe providing different views into a business web for users having different roles.

Detail Description Paragraph:
[0067] Preferred embodiments of the present invention build on this concept, and extend the role-based processing in order to provide multiple views into a business web, according to the present invention. In preferred embodiments, the specification of the role that corresponds to the user's current log-on status is stored as an attribute of the user's profile. For example, when a systems administrator logs on with his/her administrative identifier and password, these values will preferably identify a user profile where the user's role is "admin" (or some semantic equivalent). If this same person logs on with another identifier, such as a regular employee identifier, then that identifier and password preferably identify a different user profile record having a different user role. The user's profile is preferably accessed using the provisioning interface. (In alternative embodiments, the role information may be stored elsewhere, and/or may be accessed using methods provided in an interface other than the provisioning interface, including a dedicated "Roles" interface.)

Detail Description Paragraph:
[0071] The provisioning invention discussed publishing each sub-service's provisioning interface to a UDDI registry using a WSDL document, thereby enabling the joining of identity systems for (sub-)services which are dynamically integrated. The provisioning invention also stated that the provisioning interface of the aggregated service can then be created by manually or programmatically selecting from the interfaces of the sub-services comprising the aggregation, and a WSDL document may be created for this new provisioning interface and published, in a recursive manner. The present invention extends this teaching to encompass authorization-relevant attributes, and thus facilitates programmatic location of, and binding to, a role-based view into dynamically integrated distributed services. As in the provisioning invention, this functionality is provided within the context of a web services work flow, where operations are identified using WSDL documents and are invoked using SOAP messages within a work flow definition.

Detail Description Paragraph:
[0083] The user's profile information may then optionally be distributed to the selected role-specific portlet(s), as shown in Block 740. This may be useful, for example, to allow personalization of the role-specific views using preferences which may be obtained from the user's profile.

Detail Description Paragraph:
[0085] As has been demonstrated, the present invention provides advantageous techniques for providing role-specific views into aggregated web services. SOAP headers are preferably used to relay user role/profile information. The disclosed

techniques enable heterogeneous user profiles to be joined in the dynamic, run-time integration environment of web services. Open standards are leveraged. Note that while particular standards (such as WSFL, SOAP, and XLink) have been referenced when describing preferred embodiments, this is for purposes of illustrating the inventive concepts of the present invention. Alternative means for providing the analogous functionality may be used without deviating from the scope of the present invention.

CLAIMS:

4. The method according to claim 1, wherein the user role is stored in a user profile associated with the user.

Previous Doc Next Doc Go to Doc#